

### AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the subject application:

#### Listing of Claims

---

1. (Currently Amended) A method of loading a driver in a host coupled to an interconnection fabric including one or more fabric-attached I/O enclosures, comprising:
- assigning an I/O controller that is within an I/O enclosure to the host;
- before loading a driver for the I/O controller into the host, send sending a verification message to the I/O enclosure, via the interconnection fabric, to determine whether a communication path exists to the I/O controller within the I/O enclosure; and
- if the I/O enclosure responds to the verification message, then loading the driver into the host.
2. (Previously Presented) The method as claimed in claim 1, further comprising:
- alternatively, before loading the driver for the I/O controller into the host, determining whether a host-fabric adapter in the host has been initialized and connected to the interconnection fabric; and

if the host-fabric adapter in the host has been initialized and connected to the interconnection fabric, then loading the driver into the host.

3. (Cancelled)

4. (Cancelled)

5. (Previously Presented) The method as claimed in claim 1, further comprising:

initializing fabric adapters and forwarding tables before determining whether the communication path exists to the I/O controller within the I/O enclosure.

6. (Currently Amended) A method of loading a driver in a host coupled to an interconnection fabric including different hosts and I/O enclosures, comprising:

assigning a plurality of I/O controllers that are within a plurality of I/O enclosures to a plurality of hosts;

determining a list of a plurality of drivers ~~that correspond to the plurality of I/O controllers to be loaded into the plurality of hosts, the plurality of~~ drivers corresponding to the plurality of I/O controllers;

before loading the plurality of drivers into the plurality of hosts, for each of the plurality of drivers driver, sending a verification message to the

I/O controller that corresponds to the driver; and

modifying the list of plurality of drivers if a response to any of the verification messages is ~~has been~~ received.

7. (Cancelled)

8. (Currently Amended) The method as claimed in claim 6, further comprising:

determining the list of plurality of drivers, at least in part, by sending a message to a subnet manager to request a list of I/O controllers assigned to the plurality of hosts ~~a host~~.

9. (Currently Amended) The method as claimed in claim 6, further comprising:

determining the list of plurality of drivers, at least in part, by scanning the interconnection fabric for I/O controllers.

10. (Currently Amended) The method as claimed in claim 6, further comprising:

obtaining the list of plurality of drivers from a storage.

11. (Currently Amended) The method as claimed in claim 6, wherein:

receipt of the response confirms that initialization has been completed of a ~~subnet manager has finished initializing a~~ local channel adapter

port, a remote channel adapter port, and forwarding tables in intervening switches within the interconnection fabric that will be used in communication between a given one of the plurality of drivers ~~driver~~ to be loaded and a corresponding one of the plurality of I/O controllers ~~controller that corresponds to the driver~~.

12. (Currently Amended) The method as claimed in claim 6, further comprising:

notifying a fabric control driver when local channel adapter ports in a given one of the plurality of hosts ~~host~~ are configured and ready for fabric connectivity.

13. (Currently Amended) A computer readable medium having stored thereon instructions which, when executed by a processor, cause the processor to perform a method for loading a driver in a host, said method comprising:

assigning a plurality of I/O controllers that are within a plurality of I/O enclosures to a plurality of hosts;

determining a list of a plurality of drivers that correspond to the plurality of I/O controllers to be loaded into the plurality of hosts;

before loading any of the plurality of drivers into the plurality of hosts, for each of the plurality of drivers ~~driver~~, sending a verification message to a corresponding ~~the I/O controller that corresponds to~~

~~the driver~~; and

modifying the list of plurality of drivers if a response to any of the  
verification messages is ~~has been~~ received.

14. (Currently Amended) The computer readable medium as claimed in claim  
13, further comprising:

receiving an interrupt before modifying the list of plurality of drivers.

15. (Currently Amended) The computer readable medium as claimed in claim  
13, further comprising:

determining the list of plurality of drivers, at least in part, by sending a  
message to a subnet manager to request a list of I/O controllers  
assigned to the plurality of hosts ~~a host~~.

16. (Currently Amended) The computer readable medium as claimed in claim  
13, further comprising:

determining the list of plurality of drivers, at least in part, by scanning the  
interconnection fabric for I/O controllers.

17. (Currently Amended) The computer readable medium as claimed in claim  
13, further comprising:

obtaining the list of plurality of drivers from a storage.

18. (Currently Amended) The computer readable medium as claimed in claim 13, wherein:

receipt of the response confirms that initialization has been completed of a subnet manager ~~has finished~~ initializing a local channel adapter port, a remote channel adapter port, and forwarding tables in intervening switches within the interconnection fabric that will be used in communication between a given one of the plurality of drivers ~~driver~~ to be loaded and a corresponding one of the I/O controllers ~~controller that corresponds to the driver~~.

19. (Currently Amended) The computer readable medium as claimed in claim 13, further comprising:

notifying a fabric control driver when local channel adapter ports in a given one of the plurality of hosts ~~host~~ are configured and ready for fabric connectivity.

20. (Currently Amended) A network comprising:

an interconnection fabric;

a host comprising an operating system and at least one a host-fabric adapter provided to interface with the interconnection fabric; and

an I/O enclosure including at least one fabric-attached I/O controller assigned to the host and attached to the interconnection fabric;

wherein, for a given host-fabric adapter of the at least one host-fabric adapter, and for a given fabric-attached I/O controller of the at least one fabric-attached I/O controller, the operating system within the host is capable of:

determining ~~configured to determine~~ if the given host-fabric adapter has been initialized for fabric communication; and, ~~then to~~ load

if the given host-fabric adapter has been initialized for fabric communication, loading a driver that corresponds to the given fabric-attached I/O controller assigned to the host, into the host for communication with the fabric-attached I/O controller, via the interconnection fabric.

21. (Currently Amended) The network as claimed in claim 20, wherein:

the interconnection fabric includes one or more intervening switches having forwarding tables for forwarding data from the host to the fabric-attached I/O controller; and

the operating system includes a fabric control driver ~~configured~~ to provide bus abstraction and to result in ~~cause~~ the loading of the driver for the corresponding fabric-attached I/O controller, including determining whether a communication channel to the fabric-attached I/O controller exists before loading into the host the driver

that corresponds to the fabric-attached I/O controller.

22. (Currently Amended) The network as claimed in claim 21, wherein the ~~fabric control~~ driver is notified when host-fabric adapter ports are configured and ready for fabric connectivity.
23. (Cancelled)
24. (Cancelled)
25. (Currently Amended) The network as claimed in claim 20, wherein the ~~fabric control~~ driver sends ~~is configured to send~~ a verification message to the I/O enclosure via enclosure, ~~via~~ the interconnection fabric.
-